Computer Science 703

Advance Computer Architecture

2010 Semester 1

Lecture Notes

4Jun10

Review

James Goodman

Department of Computer Science

2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

Reminder:
Project is due TODAY!

2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

Test Format

- Similar in format to test
- 11 questions
- First three are compulsory
- Answer 5 out of final 8 questions

2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

Required Reading List

- Olukotun & Hammond, The Future of Microprocessors, *IEEE Queue*, September 2005.
- Hennessy & Patterson, *Computer Architecture: A Quantitative Approach* (4th Ed.), 2007, Morgan Kaufmann, San Francisco, CA, USA.
  - Chapter 2
- P. Sweazey and A.J. Smith, A class of compatible cache consistency protocols and their support by the IEEE Futurebus, *Proc. Thirteenth International Symposium on Computer Architecture (ISCA-13),* Tokyo, Japan, pp. 414-423, June 1986.
- James E. Smith, Characterizing computer performance with a single number, *Communications of the ACM*, Vol. 31, #10 (October 1988), pp 1202-1206.
- Mark Hill, Processors should support simple memory-consistency models, *IEEE Computer*, **31**(8), pp. 28-34, August 1998.
- How to Survive the Multicore Software Revolution (or at Least Survive the Hype)
- M. Herlihy and J.E.B. Moss, Transactional Memory: Architectural Support for Lock-Free Data Structures, *Proc. International Symposium on Computer Architecture (ISCA-93),* ACM Press, 1993, pp. 289-300. *(required reading for Final Exam, not Test)*
- J.R. Larus, R. Rajwar, *Transactional Memory*, (available for free download as a pdf on UoA campus) Morgan and Claypool Publishers, 2006. *(Chapters 1 & 2 are required reading for Final Exam, not Test)*
- R. Rajwar & J.R. Goodman, Speculative Lock Elision: enabling highly concurrent multithreaded execution, *34th Annual International Symposium on Microarchitecture (MICRO-34),* December 2001, pp. 294-305.

http://www.cs.auckland.ac.nz/courses/compsci703s1c/resources/

Recommended Reading

- Wikipedia: "Cache Memory" http://en.wikipedia.org/wiki/CPU_cache
- M. Hill & A.J. Smith, Evaluating Associativity in Caches, *IEEE Computer*, **29**(12), pp. 66-76, Dec 1996.
- Jon Stokes, Understanding CPU caching and performance, http://arstechnica.com/old/content/2002/07/caching.ars
- R. Rajwar & J.R. Goodman, Transactional execution: toward reliable, high-performance multithreading, *IEEE Micro,* **23**(6), pp. 117-125, November/December 2003.
- D Dice, Y Lev, M Moir, D Nussbaum, "Early experience with a commercial hardware transactional memory implementation," *ASPLOS09, pp. 157-168, (March 2009).*
- *AMD Advanced Synchronization Facility: Proposed Architectural Specfication,* Advanced Micro Devices, Publication #45432(revision 2.1), March 2009.
- M Herlihy, V Luchangco, M Moir, W Scherer III, "Software transactional memory for dynamic-sized data structures (DSTM)," *Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC),* 2003, pp. 92-101.
- M.J. Flynn, Very high-speed computing systems, *IEEE Proceedings*, **54**(12), pp. 1901-1909, 1966.
- R.M. Russell, The CRAY-I computer system, *CACM*, **21**(1), pp. 63-72, 1978.

Topics I Plan[ned] to Cover

- Moore's Law
- Multiprocessing, Multithreading & Multicores
- Classical Advanced Computer Architecture
  - Instruction-level parallelism
    - Pipelining; multiple issue & OOO execution
    - Reducing the cost of branches
    - Memory hierarchies
    - Hardware-based Speculation
  - Limits on ILP
  - Memory Hierarchy design
    - Cache memory: basics and optimizations
    - Protection: virtual memory & virtual machines
  - (I/O)
- Caches: MOESI, directories, source snooping
- Methods-tools, benchmarks, simulation
- Programming
  - Threads
  - MPI
  - OpenMP
- Caches & multiprocessors
  - coherence: snooping, MOESI, directories
  - lost updates
  - memory ordering
- Transactions & Synchronization
  - Locks, Atomic operations
  - Transactional Memory
- Hardware support for transactions
- ~~Scalable Systems~~
- ~~Dataflow~~
- SIMD/Graphics

Topics Covered (Lecture titles)

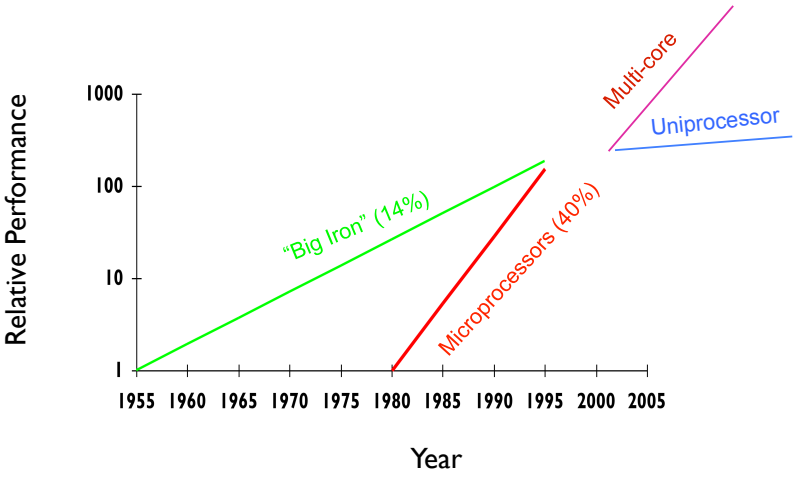
- Moore's Law
- Multiprocessing, Multithreading & Multicores
- Classical issues in computer architecture
- Concurrent Programming
- MP Cache Issues & Transactional Memory
- Caches: MOESI, directories, source snooping
- Sriram Vajapayem: Practical Multicores
- Methods-tools, simulation
- MP issues: missing update, memory ordering
- Transactional Memory: Herlihy/Moss
- SLE/TLR
- Herlihy&Moss: TM
- Extending HTM
- Non-blocking synchronization
- Software TM
- Best-effort/LogTM
- Hybrid TM
- Historyic Parallel computing (SIMD/MIMD)
- Current Parallel Computing

Terms You Should Know

- MESI, MOESI
- Bloom Filter
- STM, HTM, NZTM
- SLE
- SIMD
- ROB
- Non-blocking, lock-free, obstruction-free

High-level Summary (Slides)

Multi-cores!



The Future of Microprocessors

- Is Moore's Law really a "law"?
  - Clearly it works by creating expectations that are then fulfilled.
- How long will it last?
- Does it apply to other things than transistors?

Central Importance of Memory

- Some workloads (i.e., scientific/engineering codes) are limited by computation; most are not. The former can be parallelised easily.
- ***Latency*** of memory is critical.
- Memory accesses are highly non-random; most can be predicted to be within a small set (locality)
- Primary limit on performance: unpredicted memory accesses.

Claim: Any application that runs slow enough that performance is an issue must have massive parallelism

Multiprocessors, Multi-Cores, Multi-threading and Hyperthreading

Terminology

- Multiprocessors: multiple processors sharing a common memory (SMP, tightly-coupled MP)
- Multi-cores: multiple processors sharing a common silicon die and memory system (CMP)
- Multithreading: a single processor capable of maintaining the state of multiple threads or processes while executing
- Hyperthreading: Intel's term for a certain type of multithreading (SMT)
- Chip Multithreading (CMT): a multi-core die with multithreaded processors

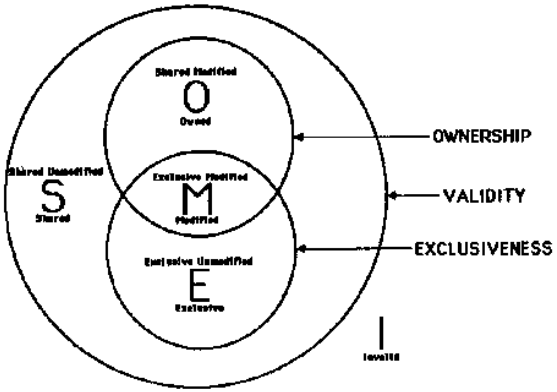
Exceeding the Flynn Limit

- Michael Flynn observed that every uniprocessor built or proposed had a maximum rate of execution of one instruction per clock cycle
- Major challenge: how to detect hazards and guarantee correctness while issuing multiple instructions simultaneously?

Memory Systems (Caches)

- Basic Ideas & Organization
  - Placement: hashing
  - Replacement: LRU; approximate LRU; random
  - Block size
  - Associativity
  - Capturing temporal/spatial locality
  - Writing
    - write-through/write-back
    - write-allocation
    - write buffers
  - Capacity, Compulsory & Conflict misses
  - Hit/Miss ratio; Avg. Memory Access Time (AMAT)
- Cache hierarchy
  - Registers, cache, main memory, disk
  - Inclusion vs. exclusion
  - Optimizations
  - Specialized (I-cache, D-cache, TLB)
  - Victim cache
- Requirements of main memory
  - Blocking/multiple requests
  - Interleaving; "false interleaving"
  - Miss Status Holding Registers MSHR
- I/O and other conflicts
- Virtual memory (latency and aliasing)

The MOESI States



2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

Evaluation

- Why?
  - Quantitative information is important
- What level?
  - Appropriate for the question
- How?
  1. Mean-value analysis/Analytical modelling
  2. Simulation
  3. Build it
- What?
  - Computer performance modelling
  - Benchmark applications

2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

What is a "good" benchmark?

- What are we trying to measure
  - Computation limited (integer or floating point?)
  - Memory limited
  - Control limited
- What is an appropriate programme?
  - Choice of programme can dramatically affect results
  - Parameters matter
    - Singly or collectively parameters can dramatically affect results

2010 YEAR | PRESENTATION | The University of Auckland | New Zealand

Multiprocessing Issues

- Cache Coherence
- "Missing Update Problem"
- Memory Consistency

Herlihy/Moss 1992

- This paper coined the term "Transactional Memory"
- Really an argument for lock-free data structures to avoid
  - Priority inversion
  - Convoying
  - Deadlock
- Argues for moving responsibility for synchronization away from the programmer
- Points out nice fit with hardware [Knight 1986]

2010 YEAR PRESENTATION The University of Auckland | New Zealand

Speculative Lock Elision

- Identify critical section (speculatively)
- Determine that contention is unlikely
- Speculatively execute critical section
- *Don't* acquire write permission on Lock
- Track external Read/Write conflicts
  - On conflict, try to acquire lock and continue
  - Declare misspeculation and execute without speculation
- When Release(Lock) is encountered, commit entire CS atomically

2010 YEAR PRESENTATION The University of Auckland | New Zealand

Topics in TM

- Herlihy/Moss: transactional cache
  - Rock
  - ASF
- Extending TM
- Non-blocking synchronisation
- Software TM
- Best-effort TM
- Hybrid TM

2010 YEAR PRESENTATION The University of Auckland | New Zealand

Past & Present Multiprocessors

- SIMD
  - Illiac IV
  - Cray-1 (vectors)
- Current multicores
  - Cell
  - Nvidia
  - Larrabee